Temporal Decomposition for Improved Unit Commitment in Power System Production Cost Modeling

Kibaek Kim, *Member, IEEE,* Audun Botterud, *Member, IEEE,* and Feng Qiu, *Senior Member, IEEE*

***Abstract*—Long-term planning in electric power systems requires simulations of unit commitment (UC) and economic dispatch (ED) for long time periods up to 20 years. Such simulations are conducted with production cost models (PCMs), which involve solving large-scale mixed-integer programming (MIP) problems with a high number of variables and constraints, because of the long planning horizon. We have developed new optimization modeling and solution techniques based on a decomposition scheme to reduce the solution time and improve the accuracy in PCMs. We propose a temporal decomposition method that solves the UC problem by systematically decoupling the long-horizon MIP problem into several subhorizon models. The decomposition is obtained by the Lagrangian relaxation of the time-coupling UC constraints such as ramping constraints and minimum uptime/downtime constraints. The key challenge with this decomposition approach is to solve several sub-MIP problems while effectively searching for dual variables in order to accelerate the convergence of the algorithm. We implement the temporal decomposition in the parallel decomposition framework DSP, which can solve the multiple subproblems in parallel on high-performance computing clusters. We also implement the branch-and-bound method on top of the decomposition in order to recover primal feasible solutions and find a primal optimal solution. Numerical results of the decomposition method are reported for the IEEE 118-bus system with up to an 168-hour time horizon.**

***Index Terms*—Production cost model, mixed-integer programming, decomposition method, parallel computing**

NOMENCLATURE

Sets:

$\mathcal{G}$	Generators
$\mathcal{G}_n$	Generators at bus n
$\mathcal{K}$	Generation cost blocks
$\mathcal{L}$	Transmission lines
$\mathcal{L}_n^+$	Transmission lines to bus n
$\mathcal{L}_n^-$	Transmission lines from bus n
$\mathcal{N}$	Buses
$\mathcal{T}$	Time periods, $= \{1, \ldots, T\}$, where T is the number of periods.

Parameters:

B_l	Susceptance of transmission line l
C_{gk}	Generation cost of generator g for generation cost block k
D_{nt}	Demand load of bus n at time t
DT_g	Minimum downtime of generator g
F_l	Power capacity of transmission line l
P_g^{max}	Maximum power generation of generator g
P_g^{min}	Minimum power generation of generator g
R_g^+	Ramp-up capacity of generator g
R_g^-	Ramp-down capacity of generator g
S_g	Startup cost of generator g
K_g	Commitment cost of generator g
UT_g	Minimum uptime of generator g
γ^+	Spinning-up reserve requirement
γ^-	Spinning-down reserve requirement
Θ_n^{min}	Minimum phase-angle of bus n
Θ_n^{max}	Maximum phase-angle of bus n

Variables:

f_{lt}	Power flow in transmission line l at time t
p_{gt}	Power generation from generator g at time t
s_{gkt}	Power generation from generator g at price block k at time t
r_{gt}^+	Reserve-up generation of generator g at time t
r_{gt}^-	Reserve-down generation of generator g at time t
u_{gt}	Commitment of generator g at time t
v_{gt}	Startup of generator g at time t
θ_{nt}	Phase-angle of bus n at time t

I. INTRODUCTION

Production cost models (PCMs) are a class of computational tools that simulate power system operations over an extended (multimonth or multiyear) time horizon. The models leverage optimization techniques to compute unit commitment (UC) and economic dispatch (ED) schedules for a power system. PCMs are the dominant approach to performing cost-benefit analyses in the electricity grid industry. System operators, utilities, generation companies, regulators, and policy analysts use PCMs for long-term planning purposes, analyzing the impacts of potential future configurations of the power system. For instance, PCMs are frequently used in renewable integration studies (e.g. [1]). However, as the power system evolves in terms of scale (e.g., the growing size of independent system operators) and structure (e.g., rapidly increasing renewables penetration rates [2], the introduction of smart grid technologies [3]), current PCMs are not adequately addressing the requirements with respect to the future power grid. For example, model resolution is currently sacrificed in order to obtain tractable run times, and the uncertainty associated

K. Kim is with the Mathematics and Computer Science Division at Argonne National Laboratory, Lemont, IL 60439 and also with the Computation Institute, University of Chicago, Chicago, IL 60637 USA.

A. Botterud and F. Qiu are with the Energy Systems Division at Argonne National Laboratory, Lemont, IL 60439.

Manuscript received XXX; revised XXX.

with renewables production is largely ignored. Consequently, PCMs increasingly do not reflect the evolving grid reality and consequently impact the accuracy of cost-benefit analyses that decision makers use to guide investment, regulations, and policy in the electric power industry.

The major challenge that hinders high-fidelity, multiscenario PCM simulation is computational tractability. In a realistic PCM simulation performed by system operators, the system may consist of several hundreds to thousands of buses and hundreds of generators. The PCM simulations could extend the individual UC optimization problem for multiple weeks. For a real system with a multiyear simulation period, weekly optimization horizon for the underlying UC problems, and hourly time resolution, the computational time is often impractical, especially when multiple scenarios are to be evaluated. A number of researchers have proposed approaches to reduce the computational burden, such as time-domain partitioning [4], and various decomposition and inexact approximations (e.g., [5]–[7]).

In essence, most PCMs today end up solving a number of deterministic multiday/week UC optimization problems in sequence, and this is where most of the computational effort is required. The UC problem is a fundamental part of power system planning and operation and is also a notoriously hard problem to solve from an optimization perspective, given the binary decision variables and intertemporal constraints involved. An extensive body of research has gone into improved solutions for the deterministic UC problem [8]. More recently, triggered by the influx of renewable energy, stochastic UC formulations have also received extensive attention in the research domain [9].

To increase the computational performance and accuracy of PCM simulations, we focus on solving the deterministic UC problem more efficiently by decomposing it into smaller time periods. The method, called *temporal decomposition*, is obtained by the Lagrangian relaxation of time-coupling constraints such as ramping capacities and minimum up/down time limits in the long-term UC problem. The key challenge with this decomposition approach is to solve several mixed-integer programming (MIP) problems while effectively searching for dual variables in order to accelerate the convergence of the algorithm. We develop a branch-and-bound method based on temporal decomposition that can solve multiple subproblems in parallel on high-performance computing (HPC) clusters. The method guarantees an optimal solution for the long-term UC problem.

The Lagrangian relaxation was first applied in [10] and has been an effective approach to unit commitment problems in different forms for more than two decades (e.g., [6], [11], [12]). In particular, a Lagrangian relaxation, similar to our temporal decomposition, has been applied in [6], where a long-term UC problem is decomposed into shorter-term UC problems by relaxing the time-coupling constraints for fuel and emission limits. However, the other coupling constraints, such as the ramping and minimum up/down time constraints, were ignored. Consequently, the decomposition approach in [6] provides only suboptimal solutions with unknown gaps.

A key task for an efficient Lagrangian relaxation method is finding good Lagrangian multipliers. Different methods have been developed for solving Lagrangian dual problems (e.g., [13]). We use a proximal bundle method in order to find the best Lagrangian dual bound. The proximal bundle method is a variant of the bundle method that outer-approximates the Lagrangian dual function by adding a set of linearly inequalities, with a proximal term in the objective function. Each iteration of the proximal bundle method either finds new dual multipliers for the subproblems or certifies the best Lagrangian dual bound.

However, such a Lagrangian relaxation method, also called *dual decomposition*, suffers from the lack of primal solution characterization and the inability to recover primal feasible solutions. To overcome issues, we additionally solve the dual of the Lagrangian dual problem for given linear inequalities generated from the dual decomposition. Note that this can be seen as Dantzig-Wolfe decomposition with column generation [14]. This provides the primal characterization of solutions that can be used to guarantee integer feasibility by a branching procedure in the branch-and-bound method.

The contributions of this paper are summarized as follows.

1) Developing a novel parallel temporal decomposition based on the Lagrangian relaxation of the time-coupling constraints in the UC problem.
2) Implementing the parallel temporal decomposition in an open-source software package `DSP`, which can run on either a desktop computer or HPC cluster.
3) Providing computational results that show significant reductions in solution time — at least by a factor of 12.

We note that the proposed decomposition scheme has potential applications in market and system operations, not only in PCM planning studies.

The rest of the paper is organized as follows. In Section II we present a MIP formulation for the UC problem considered in this paper. Section III presents the temporal decomposition that decouples the UC problem into shorter-time subproblems. We also present the dual and Dantzig-Wolfe decompositions of the problem, followed by the branch-and-price method based on the decomposition schemes. In Section IV we show computational results from the temporal decomposition for solving the long-term UC problem on the IEEE 118-bus system. The conclusions of this paper are discussed in Section V.

II. Unit Commitment Model for Production Cost Model

In this section we present a UC model formulation that is solved in a PCM simulation for an extended time horizon. The UC model is formulated as a MIP problem,

$$\min \sum_{g\in\mathcal{G}}\sum_{t\in\mathcal{T}}\left(K_g u_{gt} + S_g v_{gt} + \sum_{k\in\mathcal{K}} C_{gk} s_{gkt}\right) \tag{1a}$$

$$\text{s.t.} \sum_{l\in\mathcal{L}_n^+} f_{lt} - \sum_{l\in\mathcal{L}_n^-} f_{lt} + \sum_{g\in\mathcal{G}_n} p_{gt} = D_{nt},\ n\in\mathcal{N},\ t\in\mathcal{T}, \tag{1b}$$

$$f_{lt} = B_l\left(\theta_{nt} - \theta_{mt}\right),\ l=(m,n)\in\mathcal{L},\ t\in\mathcal{T}, \tag{1c}$$

$$p_{gt} = \sum_{k\in\mathcal{K}} s_{gkt},\ g \in \mathcal{G},\ t \in \mathcal{T}, \tag{1d}$$
$$r^-_{gt} \le p_{gt} \le r^+_{gt},\ g \in \mathcal{G},\ t \in \mathcal{T}, \tag{1e}$$
$$r^+_{gt} \le P^{max}_g u_{gt},\ g \in \mathcal{G},\ t \in \mathcal{T}, \tag{1f}$$
$$r^-_{gt} \ge P^{min}_g u_{gt},\ g \in \mathcal{G},\ t \in \mathcal{T}, \tag{1g}$$
$$\sum_{g\in\mathcal{G}} r^+_{gt} \ge (1+\gamma^+) \sum_{n\in\mathcal{N}} D_{nt},\ t \in \mathcal{T}, \tag{1h}$$
$$\sum_{g\in\mathcal{G}} r^-_{gt} \le (1-\gamma^-) \sum_{n\in\mathcal{N}} D_{nt},\ t \in \mathcal{T}, \tag{1i}$$
$$r^+_{gt} - p_{g,t-1} \le R^+_g u_{g,t-1} + P^{max}_g v_{gt},\ g \in \mathcal{G},\ t \ge 2, \tag{1j}$$
$$r^-_{gt} - p_{g,t-1} \ge -R^-_g u_{g,t-1} - P^{min}_g v_{gt},\ g \in \mathcal{G},\ t \ge 2, \tag{1k}$$
$$\sum_{q=\max\{1,t-UT_g+1\}}^{t} v_{gq} \le u_{gt},\ g \in \mathcal{G},\ t \in \mathcal{T}, \tag{1l}$$
$$\sum_{q=\max\{1,t-DT_g+1\}}^{t} w_{gq} \le 1 - u_{gt},\ g \in \mathcal{G},\ t \in \mathcal{T}, \tag{1m}$$
$$v_{gt} - w_{gt} = u_{gt} - u_{g,t-1},\ g \in \mathcal{G},\ t \ge 2, \tag{1n}$$
$$-F_l \le f_{lt} \le F_l,\ l \in \mathcal{L},\ t \in \mathcal{T}, \tag{1o}$$
$$\Theta^{min}_n \le \theta_{nt} \le \Theta^{max}_n,\ n \in \mathcal{N},\ t \in \mathcal{T}, \tag{1p}$$
$$u_{gt}, v_{gt}, w_{gt} \in \{0,1\},\ g \in \mathcal{G},\ t \in \mathcal{T}. \tag{1q}$$

The objective function (1a) of the problem is to minimize the sum of the commitment cost, the startup cost, and the generation cost. Constraint (1b) ensures the flow balance for each bus $n \in \mathcal{N}$ and time $t \in \mathcal{T}$. Constraint (1c) represents a linearized power flow equation based on Kirchhoff's law, modeling electricity transmission. Constraint (1d) splits power generation into price blocks $k \in \mathcal{K}$. Relations between reserve up/down and power generation are described by constraints (1e) – (1g); constraints (1f) and (1g) also represent on/off of each generator g at time t with specified generation capacities. Constraints (1h) and (1i) represent the spinning reserve requirements as a fraction of the total system load for each time period t. Constraints (1l) and (1m) ensure the minimum up- and downtime, respectively, for each generator. Equation (1n) describes the logic between commitment, startup, and shutdown decisions. Equations (1o) and (1p) are the bound constraints for transmission line capacity and phase angle, respectively. Commitment, startup, and shutdown decision values are restricted to binaries by (1q). Note that more constraints can be added to the model (1), such as fuel and emission limits [6].

In practice, a PCM simulation solves a set of UC models (1) in a rolling horizon basis with an overlapping period. Constraints (1j) – (1n) couple multiple time periods. Each time window of the rolling horizon sets simulation start conditions (e.g., generator status, generation, and reserve amount) for the *coupling* constraints. The length of the time window is chosen for the actual UC operations (e.g., 24- or 48-hour periods) with the overlap periods ranging from 0 to 5 days at the beginning of the horizon [4], resulting in up to 168-hour time periods for each time window. In particular, time windows with a longer overlap provide more accurate simulation solutions for PCM [4].

III. Temporal Decomposition of UC Problem

While solving a longer-term UC problem (1) is important for accurate PCM simulation, solving a sequence of UC problems poses a significant computational challenge in PCM simulation. We present a decomposition approach that accelerates the UC solution time by decoupling problem (1) into a number of subproblems with smaller time horizons. The decomposition can be obtained by relaxing the coupling constraints (1j) – (1n). We also highlight that our decomposition approach is different from the time domain decomposition [4], since our approach is an alternative to solving the UC problem by any generic MIP solver. We first define the set of subhorizon indices $\mathcal{J}$,

1) $\mathcal{T}_j \subset \mathcal{T}$ for $j \in \mathcal{J}$,
2) $\cup_{j\in\mathcal{J}} \mathcal{T}_j = \mathcal{T}$, and
3) $\mathcal{T}_i \cap \mathcal{T}_j = \emptyset$ for $i \ne j \in \mathcal{J}$,

where $\mathcal{T}_j$ is a subset of time horizon such that the indices for time periods are consecutive. Using the set $\mathcal{J}$, we rewrite the problem (1) to the following equivalent form with a set of coupling constraints and the others. We also define vectors $\mathbf{u}_j, \mathbf{v}_j, \mathbf{w}_j, \mathbf{p}_j, \mathbf{r}_j$, and $\mathbf{s}_j$, where the elements are respectively $u_{gt}, v_{gt}, w_{gt}, p_{gt}, r_{gt}$, and s_{gt} for $g \in \mathcal{G}, t \in \mathcal{T}_j$.

$$\min \sum_{j\in\mathcal{J}} \sum_{g\in\mathcal{G}} \sum_{t\in\mathcal{T}_j} \left(K_g u_{gt} + S_g v_{gt} + \sum_{k\in\mathcal{K}} C_{gk} s_{gkt} \right) \tag{2a}$$
$$\text{s.t. } r^+_{gt} - p_{g,t-1} \le R^+_g u_{g,t-1} + P^{max}_g v_{gt},\quad g \in \mathcal{G},\ t \in \mathcal{T}_j,\ t-1 \notin \mathcal{T}_j,\ j \in \mathcal{J}, \tag{2b}$$
$$r^-_{gt} - p_{g,t-1} \ge -R^-_g u_{g,t-1} - P^{min}_g v_{gt},\quad g \in \mathcal{G},\ t \in \mathcal{T}_j,\ t-1 \notin \mathcal{T}_j,\ j \in \mathcal{J}, \tag{2c}$$
$$\sum_{q=\max\{1,t-UT_g+1\}}^{t} v_{gq} \le u_{gt},\quad g \in \mathcal{G},\ t \in \mathcal{T}_j,\ t-UT_g+1 \notin \mathcal{T}_j,\ j \in \mathcal{J}, \tag{2d}$$
$$\sum_{q=\max\{1,t-DT_g+1\}}^{t} w_{gq} \le 1 - u_{gt},\quad g \in \mathcal{G},\ t \in \mathcal{T}_j,\ t-DT_g+1 \notin \mathcal{T}_j,\ j \in \mathcal{J}, \tag{2e}$$
$$v_{gt} - w_{gt} = u_{gt} - u_{g,t-1},\quad g \in \mathcal{G},\ t \in \mathcal{T}_j,\ t-1 \notin \mathcal{T}_j,\ j \in \mathcal{J}, \tag{2f}$$
$$(\mathbf{u}_j, \mathbf{v}_j, \mathbf{w}_j, \mathbf{p}_j, \mathbf{r}_j, \mathbf{s}_j) \in \mathcal{X}_j,\ j \in \mathcal{J}. \tag{2g}$$

Here constraints (2b)–(2f) couple two consecutive subhorizons, and $\mathcal{X}_j$ is the set of feasible solutions defined by all the noncoupling constraints for subhorizon j; that is,

$$\sum_{l\in\mathcal{L}^+_n} f_{lt} - \sum_{l\in\mathcal{L}^-_n} f_{lt} + \sum_{g\in\mathcal{G}_n} p_{gt} = D_{nt},\ n \in \mathcal{N},\ t \in T_j,$$
$$f_{lt} = B_l \left(\theta_{nt} - \theta_{mt}\right),\ l = (m,n) \in \mathcal{L},\ t \in \mathcal{T}_j,$$
$$p_{gt} = \sum_{k\in\mathcal{K}} s_{gkt},\ g \in \mathcal{G},\ t \in \mathcal{T}_j,$$
$$r^-_{gt} \le p_{gt} \le r^+_{gt},\ g \in \mathcal{G},\ t \in \mathcal{T}_j,$$

$$r^+_{gt} \le P^{max}_g u_{gt},\ g \in \mathcal{G},\ t \in \mathcal{T}_j,$$
$$r^-_{gt} \ge P^{min}_g u_{gt},\ g \in \mathcal{G},\ t \in \mathcal{T}_j,$$
$$\sum_{g\in\mathcal{G}} r^+_{gt} \ge (1+\gamma^+) \sum_{n\in\mathcal{N}} D_{nt},\ t \in \mathcal{T}_j,$$
$$\sum_{g\in\mathcal{G}} r^-_{gt} \le (1-\gamma^-) \sum_{n\in\mathcal{N}} D_{nt},\ t \in \mathcal{T}_j,$$
$$r^+_{gt} - p_{g,t-1} \le R^+_g u_{g,t-1} + P^{max}_g v_{gt},\ g \in \mathcal{G},\ (t-1), t \in \mathcal{T}_j,$$
$$r^-_{gt} - p_{g,t-1} \ge -R^-_g u_{g,t-1} - P^{min}_g v_{gt},\ g \in \mathcal{G},\ (t-1), t \in \mathcal{T}_j,$$
$$\sum_{q=\max\{\min\{\mathcal{T}_j\}, t-UT_g+1\}}^{t} v_{gq} \le u_{gt},\ g \in \mathcal{G},\ t \in \mathcal{T}_j,$$
$$\sum_{q=\max\{\min\{\mathcal{T}_j\}, t-DT_g+1\}}^{t} w_{gq} \le 1 - u_{gt},\ g \in \mathcal{G},\ t \in \mathcal{T}_j,$$
$$v_{gt} - w_{gt} = u_{gt} - u_{g,t-1},\ g \in \mathcal{G},\ (t-1), t \in \mathcal{T}_j,$$
$$-F_l \le f_{lt} \le F_l,\ l \in \mathcal{L},\ t \in \mathcal{T}_j,$$
$$\Theta^{min}_n \le \theta_{nt} \le \Theta^{max}_n,\ n \in \mathcal{N},\ t \in \mathcal{T}_j,$$
$$u_{gt}, v_{gt}, w_{gt} \in \{0,1\},\ g \in \mathcal{G},\ t \in T_j.$$

Before deriving the decomposition framework, we further simplify the formulation of problem (2). We define the decision variable vectors $\mathbf{x}_j$ such that $\mathbf{x}_j$ concatenate $(u_t, v_t, w_t, p_t, r_t, s_t)$ for $t \in \mathcal{T}_j$. In particular, $\mathbf{x}_j$ represents the decision variables for subhorizon j. Problem (2) can be written as

$$z := \min \sum_{j\in\mathcal{J}} \mathbf{c}_j \mathbf{x}_j \tag{3a}$$
$$\text{s.t.} \sum_{j\in\mathcal{J}} \mathbf{A}_j \mathbf{x}_j \ge \mathbf{b}, \tag{3b}$$
$$\mathbf{x}_j \in \mathcal{X}_j,\ j \in \mathcal{J}, \tag{3c}$$

where the objective coefficient vectors $\mathbf{c}_j$ are defined to represent (2a), and constraint (3b) represents constraints (2b)–(2f) that couple the subhorizons.

A. Lower Bounding from Dual Decomposition

We present the Lagrangian dual of problem (3) resulting from the Lagrangian relaxation of constraint (3b). We define the Lagrangian dual function as

$$\mathcal{L}(\lambda) := \mathbf{b}\lambda + \sum_{j\in\mathcal{J}} D_j(\lambda), \tag{4}$$

where λ is the dual variable corresponding to constraint (3b) and $D_j(\lambda)$ is defined as

$$D_j(\lambda) := \min_{\mathbf{x}_j \in conv(\mathcal{X}_j)} \left(\mathbf{c}_j - \lambda^T \mathbf{A}_j\right) \mathbf{x}_j. \tag{5}$$

The Lagrangian dual bound is obtained by solving

$$z_{LD} := \max_{\lambda \ge 0} \mathcal{L}(\lambda). \tag{6}$$

Note that $z \ge z_{LD} \ge z_{LP}$, where z_{LP} is the optimal objective value of the linear relaxation of problem (3). Problem (6) can be solved by a number of algorithms, such as the subgradient method and bundle method (see [15] and references therein).

We use a proximal bundle method (e.g., [16]) that outer-approximates the Lagrangian dual function $\mathcal{L}(\lambda)$ by adding linear inequalities with a regularization term of ℓ_2-norm in the objective function. A set of linear inequalities is added at each iteration. After adding κ sets of linear inequalities, the dual master problem (DMP) of the proximal bundle method is given by

$$\max \sum_{j\in\mathcal{J}} \mu_j + \mathbf{b}\lambda + \frac{1}{2\tau}\|\lambda - \hat{\lambda}\|_2^2 \tag{7a}$$
$$\text{s.t. } \mu_j \le D_j(\lambda^k) + (\mathbf{A}_j \mathbf{x}_j^k)^T(\lambda - \lambda^k),$$
$$j \in \mathcal{J},\ k = 1, \dots, \kappa, \tag{7b}$$
$$\lambda \ge 0, \tag{7c}$$

where τ is a positive constant and $\hat{\lambda}$ is the proximal center. Constraints (7b) are the linear inequalities that construct the outer-approximation of $\mathcal{L}(\lambda)$.

Algorithm 1 Dual Decomposition

Require: Initialize $\mathbf{c}_j, \mathbf{A}_j, \mathcal{X}_j, \mathbf{b}, \tau > 0, \hat{\lambda} \ge 0$, and $\epsilon \ge 0$. Set $\lambda^1 \leftarrow \hat{\lambda}, \kappa \leftarrow 1$, and $z_{LB} \leftarrow -\infty$.
1: **loop**
2: Solve (5) for given λ^κ and for each $j \in \mathcal{J}$.
3: Stop if $\mathcal{L}(\lambda^\kappa) - z_{LB} < \epsilon$
4: Update the best bound z_{LB}, the proximal center $\hat{\lambda}$, and weight τ.
5: Find an optimal solution $(\lambda^{\kappa+1}, \mu_j^{\kappa+1})$ of (7).
6: Set $\kappa \leftarrow \kappa + 1$.
7: **end loop**
8: **return** z_{LB} and $\mathbf{x}_j^k$ for $j \in \mathcal{J}$ and $k = 1, \dots, \kappa$.

We summarize the algorithmic steps of the proximal bundle method in Algorithm 1. The algorithm is initialized with problem data and parameters. Solving subproblems (5) at line 2 finds a new dual bound and generates linear inequalities (7b) for given λ^κ. To update $z_{LB}, \hat{\lambda}$, and τ at line 4, we follow Algorithm 2.1 and Procedure 2.2 in [16]. We solve the DMP (7) by adding the linear inequalities with the proximal parameters $\hat{\lambda}$ and τ. We repeat the steps 2 – 6 until the stopping criterion in line 3 is satisfied. Note that $z_{LD} \ge \mathcal{L}(\lambda^k) \ge \mathcal{L}(\lambda^{k-1})$ for $k = 2, \dots, \kappa$. The convergence of the algorithm depends on step 4 and is proved in [16].

B. Dantzig-Wolfe Decomposition

While effectively finding a tight dual bound z_{LD} of z, the dual decomposition does not find a primal bound of the problem (i.e., a primal feasible solution). In production cost modeling, finding a primal optimal solution is necessary for analyzing the electric grid system. Dantzig-Wolfe decomposition is a primal-dual pair of the dual decomposition, which constructs an innerestimate of the convex hull of $\mathcal{X}_j$ (denoted by $conv(\mathcal{X}_j)$) for $j \in \mathcal{J}$. We apply the decomposition to problem (3) by considering constraint (3b) only and estimating (3c). In particular, we define the restricted master problem (RMP) that considers constraint (3b) only for a limited number of solutions $\mathbf{x}_j \in conv(\mathcal{X}_j)$ for each $j \in \mathcal{J}$. Then, we use $\mathbf{x}_j^k$ from the pricing problem of the dual decomposition.

Therefore, for given $\mathbf{x}_j^k,\ j \in \mathcal{J},\ k = 1, \ldots, \kappa$, the RMP is given by

$$z_{DW} := \min \sum_{j \in \mathcal{J}} \sum_{k=1}^{\kappa} \mathbf{c}_j \mathbf{x}_j^k \alpha_j^k \tag{8a}$$

$$\text{s.t.} \sum_{j \in \mathcal{J}} \sum_{k=1}^{\kappa} \mathbf{A}_j \mathbf{x}_j^k \alpha_j^k \geq \mathbf{b}, \tag{8b}$$

$$\sum_{k=1}^{\kappa} \alpha_j^k = 1,\ j \in \mathcal{J}, \tag{8c}$$

$$\alpha_j^k \geq 0,\ j \in \mathcal{J},\ k = 1, \ldots, \kappa. \tag{8d}$$

Note that the feasible solutions to problem (3) are approximated by the convex combination of $\mathbf{x}_j^k \in conv(\mathcal{X}_j)$. The original variable solution is obtained by $\mathbf{x}_j = \sum_{k=1}^{\kappa} \mathbf{x}_j^k \alpha_j^k$ for $j \in \mathcal{J}$. In addition, the RMP is a linear programming problem. Let λ and μ_j be the dual variables corresponding to constraints (8b) and (8c), respectively.

C. Branch-and-Price Method

Recall that some of the elements in the original variable vector $\mathbf{x}_j = \sum_{k=1}^{\kappa} \mathbf{x}_j^k \alpha_j^k$ are restricted to being binaries. However, RMP does not necessarily find a binary feasible solution of the original problem (3). We apply the branch-and-bound method for ensuring a binary feasible solution by the branching procedure. Let $\hat{\alpha}_j^k$ be an optimal solution of RMP. In particular, this is called branch-and-price (BNP) method. For given fractional value of $\sum_{k=1}^{\kappa} \mathbf{x}_j^k \hat{\alpha}_j^k$ at a node of the branch-and-bound tree, the branching procedure creates two child nodes by adding the branching hyperplanes

$$\sum_{k=1}^{\kappa} \mathbf{x}_j^k \alpha_j^k \leq \lfloor \sum_{k=1}^{\kappa} \mathbf{x}_j^k \hat{\alpha}_j^k \rfloor \text{ and } \sum_{k=1}^{\kappa} \mathbf{x}_j^k \alpha_j^k \geq \lceil \sum_{k=1}^{\kappa} \mathbf{x}_j^k \hat{\alpha}_j^k \rceil \tag{9}$$

to each of the child nodes, respectively. Note that adding a branching hyperplane is equivalent to branching on a fractional variable $\mathbf{x}_j$ of the original problem (3). Then, the BNP method chooses a new node from the BNP tree and solves the node problem by using Algorithm 1.

We summarize the algorithmic steps of the branch-and-price method in Algorithm 2. In the initialization step of the algorithm, the BNP tree is initialized as an empty set `TREE` of nodes. Any BNP node `Node` $\in$ `TREE` represents the problem data for the node. For a given BNP node, Algorithm 2 solves the DMP and RMP of the node in lines 5 and 6. In line 9, adding the branching hyperplanes (9) to each of the nodes is equivalent to updating the problem data $\mathbf{A}_j$ and $\mathbf{b}$. The BNP algorithm terminates with $z = z_{UB}$ and returns the corresponding primal solution $\mathbf{x}_j = \sum_{k=1}^{\kappa} \mathbf{x}_j^k \hat{\alpha}_j^k$ for $j \in \mathcal{J}$.

Algorithm 2 Branch-and-Price Method

Require: Initialize the problem data $\mathbf{c}_j, \mathbf{A}_j, \mathcal{X}_j, \mathbf{b}$, upper bound $z_{UB} \leftarrow \infty$, and `TREE` $\leftarrow \emptyset$.

1: Create a root node (`Node`^{0}) for given $\mathbf{c}_j, \mathbf{A}_j, \mathcal{X}_j, \mathbf{b}$, and `TREE` $\leftarrow$ `TREE` $\cup$ {`Node`^{0}}.
2: **repeat**
3: Choose a node `Node` $\in$ `TREE`.
4: Update `TREE` $\leftarrow$ `TREE`$\backslash$ {`Node`}.
5: Call Algorithm 1 for solving the DMP of `Node`, which returns z_{LB} and $\mathbf{x}_j^k$.
6: Solve the RMP of `Node` that finds $\hat{\alpha}_j^k$ for given $\mathbf{x}_j^k$.
7: **if** $\sum_{k=1}^{\kappa} \mathbf{x}_j^k \hat{\alpha}_j^k$ is fractional **then**
8: Choose an original variable to branch.
9: Create two nodes (`Node`^{L} and `Node`^{R}) by adding each of (9) corresponding to the branching variable.
10: Update `TREE` $\leftarrow$ `TREE` $\cup$ {`Node`^{L}, `Node`^{R}}.
11: **else**
12: Update $z_{UB} \leftarrow \min\{z_{UB}, z_{LB}\}$.
13: **end if**
14: **until** `TREE` $= \emptyset$

IV. Computational Results

We present computational results for using the temporal decomposition method on test problems. We have implemented the temporal decomposition method in an open-source decomposition solver `DSP` [15], which can run on high-performance computing clusters in parallel via the MPI library. In addition, we have integrated a branch-and-bound method on top of the temporal decomposition method, which allows us to find an optimal solution to the original problem (1), as compared with finding lower bounds only. For the branch-and-price method (Algorithm 2), we have utilized an open-source software package `Coin-ALPS` [17] that implements a generic tree search framework. `DSP` uses a commercial optimization solver `CPLEX` (version 12.7) for solving the mixed-integer programming subproblems (5) of the temporal decomposition. We have modeled the problem (1) and its decomposition in the `Julia` script language, which can be read by `DSP`. All the computations were run on *Blues*, a 630-node computing cluster at Argonne National Laboratory. The Blues cluster has a QLogic QDR InfiniBand network, and each node has two octo-core 2.6 GHz Xeon processors and 64 GB of RAM. Note, however, that our implementation can also run on a laptop or workstation.

We use the IEEE 118-bus system with 118 buses, 54 generators, and 186 transmission lines. The system has a total generation capacity of 5,450 MW. The system is required to reserve 10% and 5% of the system load as spinning up/down reserves for the ability to increase and decrease the generation, respectively (i.e., $\gamma^+ = 0.1$ and $\gamma^- = 0.05$). We also consider three blocks of generation cost (i.e., $|\mathcal{K}| = 3$). Figure 1 shows the fluctuation of the system load profile used in our study. We use seven days of the load profile with 1-hour intervals. In particular, we use the estimated hourly load of the PJM system [18] for the dates from April 8 to April 14, 2016, which is scaled down to 10% to obtain the load profile used in our computational study. The load is 2,775 MW on average, with a peak of 3,182 MW.

Table I presents the size of the problem instances for 24-, 48-, 72-, 96-, 120-, 144-, and 168-hour horizons. Recall that the 168-horizon UC problem represents the 24-hour operation horizon with 6 days of overlap (lookahead) for the PCM simulation. The UC problem is solved up to a week ahead in their operational decision processes to account for generating

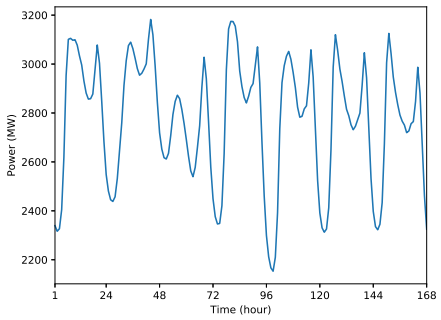


Fig. 1. Load profile used for the IEEE 118-bus system.

TABLE I
SIZES OF IEEE 118-BUS SYSTEM PROBLEM INSTANCES

T	# Constraints	# Variables	# Binary
24	19765	18960	1296
48	40070	37920	2592
72	60398	56880	3888
96	80726	75840	5184
120	101054	94800	6480
144	121382	113760	7776
168	141710	132720	9072

TABLE II
NUMERICAL RESULTS FOR PROBLEM INSTANCES USING CPLEX-12.7 IN PARALLEL WITH 16 COMPUTING CORES

T	Best Objective	Gap (%)	Time (sec.)
24	1077030.3	0	6
48	2171642.3	0	68
72	3122813.6	0	1351
96	4174770.7	< 0.01	14400
120	5158594.4	< 0.01	14400
144	6152020.7	0.01	14400
168	7122822.5	0.02	14400

units with very long startup times [19]. Table II summarizes the numerical results for each problem instance solved by CPLEX in parallel on a 16-core single node of the Blues cluster. We set a zero optimality gap tolerance and a 4-hour wall clock time limit. Within the time limit, optimal objective values were found for 24-, 48-, and 72-hour problem instances. For the other problem instances, CPLEX found solutions with positive gaps. In particular, the optimality gap increases as the problem size increases, as shown in Table II.

We now present the numerical results from the temporal decomposition. We tested the decomposition method with different numbers of subhorizons (i.e., $|\mathcal{J}| = 2, 4, 8, 12, 24$); Table III shows the size of the coupling problem (2) that results. The percentages of the coupling constraints and variables to the total numbers are also reported in the table. The number of constraints and variables increases with the number of decompositions. However, we highlight that the temporal decomposition generates columns iteratively up to the number of coupling variables reported in Table III. Therefore, the size of the master problem is far smaller than the size of the coupling problem.

In Figure 2 and Table IV, we report numerical results from the temporal decomposition method with different numbers of

TABLE III
NUMBER OF COUPLING CONSTRAINTS AND VARIABLES

T	$\lvert\mathcal{J}\rvert$	Constraints	Variables	Binary
24	2	594 (3%)	3456 (18%)	702 (54%)
	4	1506	4104	1026
	8	2548	4914	1188
	12	3300	5616	1242
	24	5428 (27%)	7614 (40%)	1296 (100%)
48	2	617 (1%)	6588 (17%)	1350 (52%)
	4	1782	7668	1998
	8	3514	8640	2322
	12	4510	9396	2430
	24	6900 (17%)	11448 (30%)	2538 (97%)
72	2	629 (1%)	8532 (15%)	1998 (51%)
	4	1818	11232	2970
	8	4032	12366	3456
	12	5522	13176	3618
	24	8372 (13%)	15282 (26%)	3780 (97%)
96	2	640 (0.7%)	10314 (13%)	2592 (50%)
	4	1851	14688	3942
	8	4158	16092	4590
	12	6182	16956	4806
	24	9430 (11%)	19116 (25%)	5022 (96%)
120	2	640 (0.6%)	10314 (10%)	2592 (40%)
	4	1869	17604	4914
	8	4200	19818	5724
	12	6424	20736	5994
	24	10488 (10%)	22950 (24%)	6264 (96%)
144	2	640 (0.5%)	10314 (9%)	2592 (33%)
	4	1887	20520	5886
	8	4242	23544	6858
	12	6534	24516	7182
	24	11546 (9%)	26784 (20%)	7506 (96%)
168	2	640 (0.4%)	10314 (7%)	2592 (28%)
	4	1905	23436	6858
	8	4284	27270	7992
	12	6578	28296	8370
	24	12604 (8%)	30618 (23%)	8748 (96%)

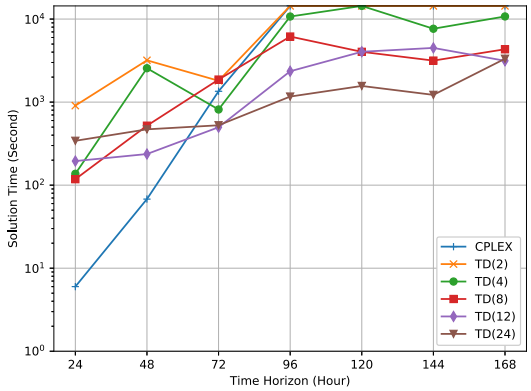


Fig. 2. Solution times resulting from the temporal decompositions for different numbers of time periods.

subintervals for 24-, 48- ,72-, 96-, 120-, 144-, and 168-horizon UC problems. The temporal decomposition (Algorithm 2) solves the MIP pricing subproblem (5) for each $j \in \mathcal{J}$ in parallel. Each subproblem is solved on a single node with 16 cores. For example, the problem of $|\mathcal{J}| = 24$ solves 24 subproblems on 24 computing nodes with 384 (=24*16) cores. The quadratic programming master problem is also solved in parallel by CPLEX on a single node with 16 cores. We also set 4-hour time limit for the runs.

Figure 2 plots the solution times resulting from CPLEX

TABLE IV
NUMERICAL RESULTS FROM TEMPORAL DECOMPOSITION WITH DIFFERENT NUMBER OF SUB-HORIZONS.

		Root Node				Branch-and-Price				
T	$\lvert\mathcal{J}\rvert$	Iterations	Best Bound	Gap (%)	Time (sec.)	Nodes	Best Bound	Best Objective	Gap (%)	Time (sec.)
24	2	17	1077030.3	0	908	1	1077030.3	1077030.3	0	908
	4	54	1077030.3	0	137	1	1077030.3	1077030.3	0	137
	8	92	1077023.6	< 0.01	103	5	1077030.3	1077030.3	0	118
	12	120	1077030.3	0	148	13	1077030.3	1077030.3	0	195
	24	140	1077023.0	< 0.01	237	13	1077030.3	1077030.3	0	342
48	2	49	2171629.6	< 0.01	2092	11	2171642.3	2171642.3	0	3187
	4	71	2171623.1	< 0.01	2526	3	2171642.3	2171642.3	0	2559
	8	90	2171620.9	< 0.01	435	5	2171642.3	2171642.3	0	520
	12	105	2171622.2	< 0.01	176	13	2171642.3	2171642.3	0	237
	24	129	2171635.1	< 0.01	237	35	2171642.3	2171642.3	0	470
72	2	32	3122813.6	0	1941	1	3122813.6	3122813.6	0	1807
	4	37	3122800.0	< 0.01	858	11	3122813.6	3122813.6	0	815
	8	65	3122792.5	< 0.01	1289	11	3122813.6	3122813.6	0	1868
	12	87	3122810.3	< 0.01	436	19	3122813.6	3122813.6	0	498
	24	129	3122806.5	< 0.01	299	15	3122813.6	3122813.6	0	526
96	2	48	4174726.5	< 0.01	14400	0	4174726.5	NA	< 0.01	14400
	4	47	4174746.5	< 0.01	3803	19	4174770.7	4174770.7	0	10730
	8	84	4174763.3	< 0.01	4247	35	4174770.7	4174770.7	0	6160
	12	112	4174747.4	< 0.01	1473	33	4174770.7	4174770.7	0	2352
	24	141	4174769.4	< 0.01	735	29	4174770.7	4174770.7	0	1166
120	2	48	5158553.5	< 0.01	14400	0	5158553.5	NA	< 0.01	14400
	4	55	5144130.4	0.28	14400	0	5144130.4	NA	0.28	14400
	8	99	5158562.0	< 0.01	3148	5	5158594.4	5158594.4	0	4033
	12	116	5158568.2	< 0.01	3355	15	5158594.4	5158594.4	0	4039
	24	129	5158570.9	< 0.01	829	35	5158594.4	5158594.4	0	1565
144	2	47	6150192.3	0.03	14400	0	6150192.3	NA	0.03	14400
	4	65	6151935.2	< 0.01	7108	3	6151974.0	6151974.0	0	7646
	8	77	6151928.5	< 0.01	1967	15	6151974.0	6151974.0	0	3161
	12	96	6151934.4	< 0.01	3560	19	6151974.0	6151974.0	0	4494
	24	139	6151933.5	< 0.01	1057	7	6151974.0	6151974.0	0	1222
168	2	47	7119965.0	0.04	14400	0	7119965.0	NA	0.04	14400
	4	57	7122821.1	< 0.01	10573	1	7122821.1	7122821.1	0	10753
	8	98	7122771.9	< 0.01	3600	0	7122821.1	7122821.1	0	4329
	12	108	7122773.7	< 0.01	2893	5	7122821.1	7122821.1	0	3145
	24	151	7122793.8	< 0.01	2396	15	7122821.1	7122821.1	0	3327

and the temporal decomposition method with different numbers of subintervals. The x-axis presents the time horizon of the UC problem. "TD(n)" plots the solution time from the temporal decomposition of n subintervals for $n = 2, 4, 8, 12, 24$. The temporal decomposition method found optimal solutions for all the problem instances (i.e., $T = 24, 48, 72, 96, 120, 144, 168$) within the time limit. In particular, the 96-hour horizon problem was solved to optimality after 1166 seconds when decomposed into 24 subhorizons. For this problem instance, the solution time was reduced by at least a factor of 12.

Detailed numerical results are reported in Table IV. The columns for "Root Node" present the results observed at the root node before starting the branch-and-price method. The column "Iterations" reports the number of iterations taken in Algorithm 1. The best lower bound of z are reported in the column "Best Bound" with the relative gap as the relative difference between the best bound and the best objective found in Table II. For the branch-and-price results in Table IV, "Nodes" reports the number of BNP nodes solved. "Best Objective" reports the objective value of the primal solution. The temporal decomposition found the optimal solutions and the best objective values for all the problem instances. Note, however, that the best objective values were not found within the time limit (denoted by "NA") when the subproblems were large (e.g., the 96-horizon instance with $|\mathcal{J}| = 2$).

We highlight that the best bound found at root node is very tight, with the gap less than 0.01% for most problem instances, including zero gaps for four of the instances. As a result, only a few of the BNP nodes were solved to find a primal solution and prove optimality. Note, however, that the computational performance depends on the choice of the number of subhorizons. If the size of the subhorizon problem is large, a primal solution cannot be found (e.g., the problem with $T = 168$ and $|\mathcal{J}| = 2$).

We also highlight that unit commitment decisions are considerably different, closing the 0.02% optimality gap for the 168-hour planning instance. Figure 3 plots the commitment schedules obtained by `CPLEX` and our temporal decomposition for the 168-hour time horizon instance. Specifically, the commitment schedules are different in 9 of the 54 generating units (17%) for 372 hours. These results suggest that suboptimal schedules (even with small optimality gap) can deviate significantly from an optimal schedule, thus hindering high-fidelity PCM simulations.

V. CONCLUSIONS

We presented a novel temporal decomposition method that solves a long-term UC problem by splitting the long-term problem into many shorter-term problems based on Lagrangian relaxation, where the shorter-term problems are solved in parallel. The computational results on the IEEE 118-bus system showed that the parallel temporal decomposition reduces the

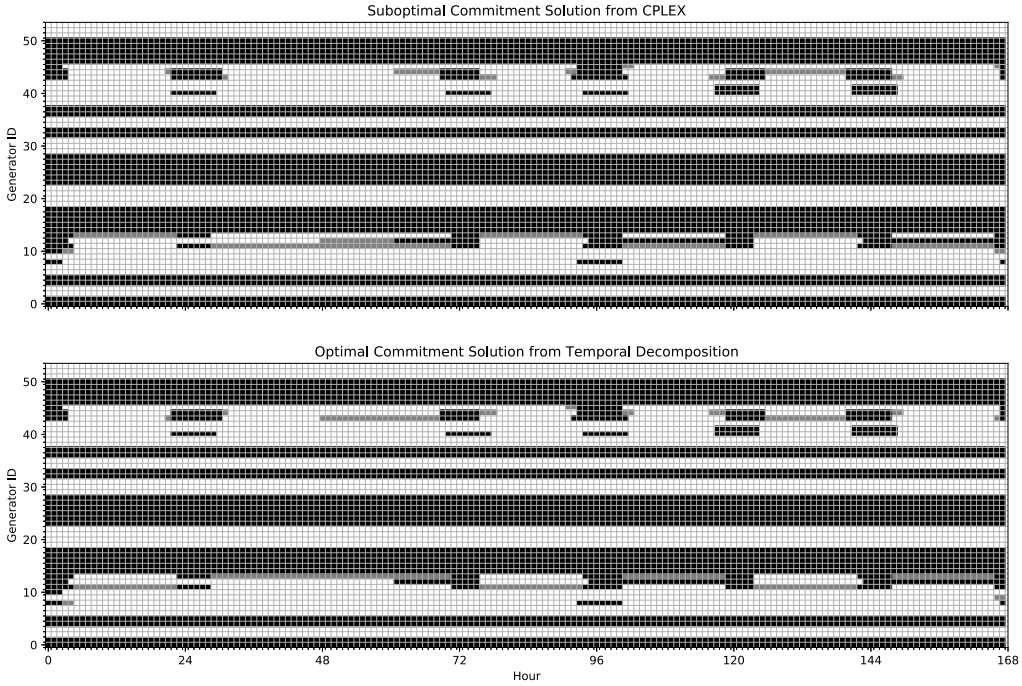


Fig. 3. Unit commitment solutions found for the 168-hour unit commitment problem instance by CPLEX and the temporal decomposition TD(24). The generators are scheduled online for the black-colored time periods with the grey highlights of differences.

solution time by 92% for solving a 96-hour horizon UC problem, which could not be solved to optimality by CPLEX with the 4-hour time limit. The improvement in solution quality and time allows more efficient high-fidelity PCM simulations with multiple scenarios. We highlight that our decomposition scheme can be applied to other problems such as market and system operations. We plan to develop and integrate primal methods (e.g., cutting planes and heuristics) in the temporal decomposition, which would accelerate the solution.

Acknowledgment

This material is based upon work supported by the U.S. Department of Energy, Office of Electricity Delivery & Energy Reliability/Office of Energy Efficiency & Renewable Energy, under contract number DE-AC02-06CH11357. We gratefully acknowledge the computing resources provided on Blues, a high-performance computing cluster operated by the Laboratory Computing Resource Center at Argonne National Laboratory.

References

[1] A. Bloom, A. Townsend, D. Palchak, J. King, E. Ibanez, C. Barrows, M. Hummon, and C. Draxl, "Eastern renewable generation integration study," National Renewable Energy Laboratory, Tech. Rep. NREL/TP-6A20-64472, 2016.

[2] REN21, *Renewables 2016: Global Status Report*. Renewable Energy Policy Network for the 21st Century (REN21), 2016.

[3] H. Farhangi, "The path of the smart grid," *IEEE Power and Energy magazine*, vol. 8, no. 1, 2010.

[4] C. Barrows, M. Hummon, W. Jones, and E. Hale, "Time domain partitioning of electricity production cost simulations," National Renewable Energy Laboratory, Tech. Rep. NREL/TP-6A20-60969, January 2015.

[5] H. Ma and S. Shahidehpour, "Transmission-constrained unit commitment based on Benders decomposition," *International Journal of Electrical Power & Energy Systems*, vol. 20, no. 4, pp. 287–294, 1998.

[6] Y. Fu, M. Shahidehpour, and Z. Li, "Long-term security-constrained unit commitment: hybrid dantzig-wolfe decomposition and subgradient approach," *IEEE Transactions on Power Systems*, vol. 20, no. 4, pp. 2093–2106, 2005.

[7] K. Kim and V. M. Zavala, "Large-scale stochastic mixed-integer programming algorithms for power generation scheduling," in *Alternative Energy Sources and Technologies*. Springer, 2016, pp. 493–512.

[8] N. P. Padhy, "Unit commitment-a bibliographical survey," *IEEE Transactions on Power Systems*, vol. 19, no. 2, pp. 1196–1205, 2004.

[9] Z. Zhou, C. Liu, and A. Botterud, "Stochastic methods applied to power system operations with renewable energy: A review," Argonne National Laboratory, Tech. Rep. ANL/ESD-16/14, 2016.

[10] J. A. Muckstadt and S. A. Koenig, "An application of Lagrangian relaxation to scheduling in power-generation systems," *Operations Research*, vol. 25, no. 3, pp. 387–403, 1977.

[11] W. L. Peterson and S. R. Brammer, "A capacity based Lagrangian relaxation unit commitment with ramp rate constraints," *IEEE Transactions on Power Systems*, vol. 10, no. 2, pp. 1077–1084, 1995.

[12] T. Li and M. Shahidehpour, "Price-based unit commitment: A case of lagrangian relaxation versus mixed integer programming," *IEEE transactions on power systems*, vol. 20, no. 4, pp. 2015–2025, 2005.

[13] N. J. Redondo and A. Conejo, "Short-term hydro-thermal coordination by Lagrangian relaxation: solution of the dual problem," *IEEE Transactions on Power Systems*, vol. 14, no. 1, pp. 89–95, 1999.

[14] G. B. Dantzig and P. Wolfe, "Decomposition principle for linear programs," *Operations Research*, vol. 8, no. 1, pp. 101–111, 1960.

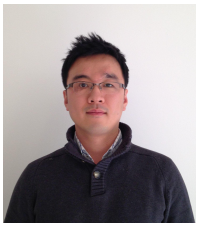
[15] K. Kim and V. M. Zavala, "Algorithmic innovations and software for the dual decomposition method applied to stochastic mixed-integer programs," *Mathematical Programming Computation (to appear)*, 2017.

[16] K. C. Kiwiel, "Proximity control in bundle methods for convex nondifferentiable minimization," *Mathematical Programming*, vol. 46, no. 1, pp. 105–122, 1990.

[17] Y. Xu, T. K. Ralphs, L. Ladányi, and M. J. Saltzman, "Alps: A framework for implementing parallel tree search algorithms," in *The Next Wave in Computing, Optimization, and Decision Technologies*. Springer, 2005, pp. 319–334.

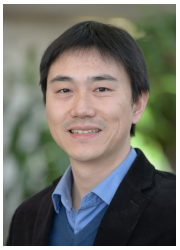
[18] "PJM - Estimated Hourly Load," http://www.pjm.com/markets-and-operations/energy/real-time/loadhryr.aspx, accessed: 2017-04-26.

[19] J. N. Puga, "The importance of combined cycle generating plants in integrating large levels of wind power generation," *The Electricity Journal*, vol. 23, no. 7, pp. 33–44, 2010.



Kibaek Kim Kibaek Kim is an assistant computational mathematician in the Mathematics and Computer Science Division at Argonne National Laboratory. He holds Ph.D. and M.S. degrees from Northwestern University and a B.S. degree from Inha University in Korea, all in industrial engineering. He currently serves as a reviewer in several peer-reviewed journals, including Operations Research, Mathematical Programming, Computational Optimization and Applications, European Journal of Operational Research, and IEEE Transactions on Power Systems. His research interests are in modeling and parallel algorithms for large-scale optimization problems in application to network design, planning, and operations. He is a main developer of DSP, the parallel structured mixed-integer programming solver used for the computational studies in this paper.

Audun Botterud Audun Botterud received his M.Sc. degree in industrial engineering and Ph.D. degree in electrical power engineering from the Norwegian University of Science and Technology, Trondheim, Norway, in 1997 and 2004, respectively. He is a principal energy systems engineer in the Energy Systems Division at Argonne National Laboratory and a principal research scientist in the Laboratory for Information and Decision Systems at the Massachusetts Institute of Technology. He was previously with SINTEF Energy Research in Trondheim, Norway. His research interests include power systems planning and economics, electricity markets, grid integration of renewable energy, energy storage, and stochastic optimization. Dr. Botterud is cochair for the IEEE Task Force on Bulk Power System Operations with Variable Generation.



Feng Qui Feng Qiu received his Ph.D. from the School of Industrial and Systems Engineering at the Georgia Institute of Technology in 2013. He is a computational scientist with the Energy Systems Division at Argonne National Laboratory. His current research interests include optimization in power system operations, electricity markets, and power grid resilience.

This material is based upon work supported by the U.S. Department of Energy, Office of Science, under contract number DE-AC02-06CH11357.